

RELATED APPLICATIONS

5

10

15

20

25

Unfortunately, this has resulted in a continually increasing number of messages being generated from applications and the operating system. Additionally, the number of

5 Dialog boxes, error messages, message boxes and other notification windows form all of these sources are invasive and often frustrate the user. These boxes may appear to the user as the top-most window and they generally capture control of the user interface. These dialog and message boxes are often accidentally dismissed by users who may be focusing on entering input and not paying attention to the computer monitor.

Tray icon notifications, on the other hand, provide an indication on the status bar that a message has arrived. This indication is generally displayed using one of two methods. In the first method, the tray icon provides notification that a message has been received by either changing its shape, blinking, or a combination of the two. The second method places a tray icon on the status bar indicating that a message has been received. This second method then removes the tray icon when the user has seen the message. Two significant limitations of using tray icons are that they are not always obvious when they appear and the tray icon is not always easily identifiable. Another limitation is that the tray icon notification is not practicable in those operating systems that provide the user

with an option to hide the status bar because the user will not see the tray icon notification. Additionally, when multiple messages are present, the tray notify icon area can become cluttered making new icons hard to spot. As a result of the clutter, the user is often unaware when a new message arrives.

5 In response to these and other user frustrations with dialog boxes, message boxes and tray icons, application developers have developed individualized notification mechanisms for their applications in an attempt to reduce the user frustration. For example, many e-mail applications now provide some form of message indicating the receipt or presence of e-mail. When a user is logged on to MSN® Messenger Service, for
10 example, a message window appears to inform the user whenever mail arrives. However, control of the user interface is not transferred to this message window. WorldNet® provides a tray icon that blinks when a message arrives and displays a message to indicate the number of messages that have not been read when the user places a cursor over the tray icon.

15 The development of independent notification mechanisms provide some advantages over prior methods of providing user notification. However, the mechanisms have resulted in inconsistent user interfaces for managing notifications and have introduced other limitations. Each application providing an independent mechanism requires additional overhead. Further, the effort of coding and debugging these individual
20 notification mechanisms and their user interfaces has increased development time and cost of these applications. Unfortunately, users still cannot control how notifications are received and cannot disable the notifications from being received. What is needed is a

09705656.110300

notification mechanism that is shared between all applications and that provides users with more control over the notifications they receive.

SUMMARY OF THE INVENTION

5 In view of the above described problems existing in the art, the present invention provides a centralized notification component that may be shared between all applications. Through this centralized component users are provided the capability to manage their own notifications locally. Further, application developers no longer have to code and debug individual notification mechanisms and user interfaces thus reducing
10 development cycle time and cost.

The instant invention provides a client-side shared notification component that lets a user continue working on a task while receiving a notification (i.e., a message). The notification component further allows the user to indicate what notifications the user want to sees and how he would like to see them, i.e. the rendering type. Rendering types
15 include audible, visual, and pager styles. An application uses an API (Application Programming Interface) to send a notification to the notification component using schema based or text based notifications. The notification component parses the notification and provides the user the notification through notification classifications.

The notification component also allows the user to globally set preferences as to
20 how the notifications should be rendered. These preferences include the enabling or disabling of notification classifications, the selecting of which rendering type the user wants to use for each notification classification, and the setting of priority of the notifications within each notification classification.

09705858 " 110300

Additional features and advantages of the invention will be made apparent from the following detailed description of illustrative embodiments that proceeds with reference to the accompanying figures.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The file of this patent contains at least one drawing executed in color. Copies of this patent with color drawings will be provided by the Patent and Trademark Office upon request and payment of the necessary fee. While the appended claims set forth the features of the present invention with particularity, the invention, together with its objects and advantages, may be best understood from the following detailed description taken in conjunction with the accompanying drawings of which:

10

Figure 1 is a block diagram generally illustrating an exemplary computer system on which the present invention resides;

15

Figure 2 is a block diagram of a notification component of the present invention in relation to components in the exemplary computer system of Figure 1;

Figure 3 is an exemplary illustration of a user preference dialog box in accordance with the teaching of the present invention;

Figure 4 is a diagram illustrating display areas in accordance with the teaching of the present invention; and

20

Figure 5 shows illustrative visual rendering styles in accordance with the teaching of the present invention.

09705658-110300

DETAILED DESCRIPTION OF THE INVENTION

Turning to the drawings, wherein like reference numerals refer to like elements, the invention is illustrated as being implemented in a suitable computing environment. Although not required, the invention will be described in the general context of computer-executable instructions, such as program modules, being executed by a personal computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Moreover, those skilled in the art will appreciate that the invention may be practiced with other computer system configurations, including hand-held devices, multi-processor systems, microprocessor based or programmable consumer electronics, network PCs, minicomputers, mainframe computers, and the like. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote memory storage devices.

Figure 1 illustrates an example of a suitable computing system environment 100 on which the invention may be implemented. The computing system environment 100 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Neither should the computing environment 100 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 100.

The invention is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well known computing systems, environments, and/or configurations that may be suitable for use with the invention include, but are not limited to, personal computers, server computers, handheld or laptop devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

The invention may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. The invention may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

With reference to Figure 1, an exemplary system for implementing the invention includes a general purpose computing device in the form of a computer 110. Components of computer 110 may include, but are not limited to, a processing unit 120, a system memory 130, and a system bus 121 that couples various system components including the system memory to the processing unit 120. The system bus 121 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of

09705858 110300

example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Associate (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus.

- 5 Computer 110 typically includes a variety of computer readable media. Computer readable media can be any available media that can be accessed by computer 110 and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer readable media may comprise computer storage media and communication media. Computer storage media includes both volatile
- 10 and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes,
- 15 magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store the desired information and which can be accessed by computer 110. Communication media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media.
- 20 The term "modulated data signal" means a signal that has one or more of its characteristics set or changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as a wired network or direct-wired connection, and wireless media such as acoustic, RF,

09705858 "110300

infrared and other wireless media. Combinations of the any of the above should also be included within the scope of computer readable media.

09705658-110300
The system memory 130 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 131 and random access
5 memory (RAM) 132. A basic input/output system 133 (BIOS), containing the basic routines that help to transfer information between elements within computer 110, such as during start-up, is typically stored in ROM 131. RAM 132 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by processing unit 120. By way of example, and not limitation, Figure 1 illustrates
10 operating system 134, application programs 135, other program modules 136, and program data 137.

The computer 110 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Figure 1 illustrates a hard disk drive 140 that reads from or writes to non-removable, nonvolatile
15 magnetic media, a magnetic disk drive 151 that reads from or writes to a removable, nonvolatile magnetic disk 152, and an optical disk drive 155 that reads from or writes to a removable, nonvolatile optical disk 156 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that can be used in the exemplary operating environment include, but are not limited to, magnetic tape
20 cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 141 is typically connected to the system bus 121 through a non-removable memory interface such as interface 140, and

magnetic disk drive 151 and optical disk drive 155 are typically connected to the system bus 121 by a removable memory interface, such as interface 150.

The drives and their associated computer storage media discussed above and illustrated in Figure 1, provide storage of computer readable instructions, data structures, program modules and other data for the computer 110. In Figure 1, for example, hard disk drive 141 is illustrated as storing operating system 144, application programs 145, other program modules 146, and program data 147. Note that these components can either be the same as or different from operating system 134, application programs 135, other program modules 136, and program data 137. Operating system 144, application programs 145, other program modules 146, and program data 147 are given different numbers hereto illustrate that, at a minimum, they are different copies. A user may enter commands and information into the computer 20 through input devices such as a keyboard 162 and pointing device 161, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone, joystick, game pad, satellite dish, scanner, or the like. These and other input devices are often connected to the processing unit 120 through a user input interface 160 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). A monitor 191 or other type of display device is also connected to the system bus 121 via an interface, such as a video interface 190. In addition to the monitor, computers may also include other peripheral output devices such as speakers 197 and printer 196, which may be connected through a output peripheral interface 190.

The computer 110 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 180. The remote computer 180 may be another personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the personal computer 110, although only a memory storage device 181 has been illustrated in Figure 1. The logical connections depicted in Figure 1 include a local area network (LAN) 171 and a wide area network (WAN) 173, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

When used in a LAN networking environment, the personal computer 110 is connected to the LAN 171 through a network interface or adapter 170. When used in a WAN networking environment, the computer 110 typically includes a modem 172 or other means for establishing communications over the WAN 173, such as the Internet. The modem 172, which may be internal or external, may be connected to the system bus 121 via the user input interface 160, or other appropriate mechanism. In a networked environment, program modules depicted relative to the personal computer 110, or portions thereof, may be stored in the remote memory storage device. By way of example, and not limitation, Figure 1 illustrates remote application programs 185 as residing on memory device 181. It will be appreciated that the network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

In the description that follows, the invention will be described with reference to acts and symbolic representations of operations that are performed by one or more

09705858.110300

computers, unless indicated otherwise. As such, it will be understood that such acts and operations, which are at times referred to as being computer-executed, include the manipulation by the processing unit of the computer of electrical signals representing data in a structured form. This manipulation transforms the data or maintains it at locations in the memory system of the computer, which reconfigures or otherwise alters the operation of the computer in a manner well understood by those skilled in the art. The data structures where data is maintained are physical locations of the memory that have particular properties defined by the format of the data. However, while the invention is being described in the foregoing context, it is not meant to be limiting as those of skill in the art will appreciate that various of the acts and operation described hereinafter may also be implemented in hardware.

Figure 2 illustrates an embodiment of the instant invention in the exemplary computer system of figure 1, although those skilled in the art will recognize that the functions of the invention can be implemented in other computer systems. Operating system 134, application programs 135, other program modules 136, operating system 144, application programs 145, other program modules 146, and remote application programs 185 may each send a notification to be provided to the user to notification component 138. However, for purposes of illustration and ease of understanding, the invention will be described in relation to only the notifications provided by the application programs 135. Those skilled in the art will understand that the operating system 134, other program modules 136, operating system 144, application programs 145, other program modules 146, and remote application programs 185 utilize a similar mechanism to provide their notifications as application programs 135. While figure 2

shows notification component 138 being separate from system memory 130, it may be implemented as part of system memory 130. For example, notification component 138 may be integrated with operating system 134.

Notification component 138 receives the notification and checks the user's preferences stored in a system registry or file to determine if the user wants to see the notification. The user's preferences may also be stored within the notification component 138. If the user's preferences indicate that the notification should be displayed, notification component 138 renders the notification according to the user's preferences via an appropriate notification medium such as monitor 191, speakers 193, pager 198, or other output 199. Other output 199 represents any mechanism that provides notification to the user such as a wireless PDA (personal digital assistant). The notification is rendered using a rendering style that does not transfer control of the input devices to the notification and that is not hidden behind other windows.

There are several types of notifications that are rendered by notification component 138. The types of notifications include mail messages, user messages, applications changing state messages, meeting reminders, stock tickers, etc. The notification component assigns each notification type a classification (i.e., a category). The list of classifications is extensible and presently includes contact, financial, e-mail, system level, and audio classifications. The contact classification is used for meeting notices, phone notifications, and any other notification that indicates that another user wants to make contact. The financial classification is used for financial based notifications, and the audio classification is used for audio notifications. The e-mail classification is used for e-mail notifications. For example, contact classification

09705858.110300

notifications are notifications such as "Conference Room 10 in 5 minutes" for a meeting notice, and "Joe at 703-308-4357" for a phone notification. An example of an e-mail notification is "Bill from Microsoft." An example of a financial classification notification is a stock ticker such as "MSFT up 5," an example of a system level notification is

5 "Battery is low" and an example of an audio classification notification is "Playing Train."

The notification component 138 receives notifications to be rendered via an application programming interface (API) 139. In one embodiment, the application programs 135 specify the classification and the notification to be rendered in text form to the notification component 138 via the API 139. The API 139 comprises a property

10 command that identifies the classification and a text command that provides the text of the notification. An exemplary implementation of the property command and the text command is the following:

```
CTunaclient notify;  
  
    notify.SetProperty("classification")  
  
    notify.SetText("message text")
```

15

where "classification" is the notification classification and "message text" is the text of the notification to be rendered. In this embodiment, the notification component 138 receives the notification and renders it according to the user's preferences.

In an alternative embodiment, the application programs 135 provide the

20 notification to the notification component 138 using an XML (extensible markup language) schema. The XML schema is extensible and uses a notice classification tag and a notification type tag. The notice classification tag provides the notification component 138 with the notification classification. The notification type tag provides the

09705858.110300

notification component 138 with the notification to be rendered and the style of rendering. An exemplary implementation of the XML schema is the following:

```
<notification classification="category">
  <style text="message"/>
5  </notification>
```

where "category" is the notification classification, "message" is the notification to be rendered and "style" is the style of rendering. The rendering styles include visual, pager, and spoken renderings. For example, if the notification is to be rendered visually, the <style text="message"> command is <gui text="message">. For audio notifications, the 10 command is <spoken text="message"> and for pager notifications, the command is <pager text="message">. The applications programs 135 can supply several text strings to the notification component 138. An example of a text string the application programs 135 provides is:

```
<notification>
15  <gui text="Message"/>
    <pager text="You have a message"/>
    <spoken text="There is a message for you"/>
  </notification>
```

The notification component 138 then decides which rendering style to use based on the 20 user's preferences.

The notification component 138 renders the notifications in accordance with the notification classification and the user's preferences. The user preferences allow the user to enable or disable notification classifications globally, to enable or disable notifications

09705858.110300

from a particular application, to specify the rendering style for each notification classification, and to specify the priority in which notifications should be rendered. In one embodiment, users select their preferences via a dialog box.

Figures 3a and 3b provide an example of a dialog box that is used to set user preferences. Dialog box 200 has several preference screens represented by tabs 202 and 204. The user selects tab 202 to bring up the global preferences and selects tab 204 to bring up the notification classifications. The global preferences allows the user to enable the audio rendering style by selecting the check box 208, which corresponds to the description 206. Likewise, the pre-notification notification, discussed below, can be enabled by selecting the check box 201, which corresponds to the description 212. The notification classification rendering preferences are selected from list box 216. The user selects how notifications in each notification classification are rendered by selecting check box 218 for visual rendering and by selecting check box 220 for audio rendering. The user can select the font and font size of the visual rendering style by selecting font button 222, which bring ups another dialog box (not shown) that allows the user to select font options. One skilled in the art will recognize that other tabs (not shown) are selected to bring up other preference screens. For example, in one embodiment, the functional notification classification provides the user with an option to have the notification component 138 bring up the web-site where the user's portfolio is stored when the user receives notifications that affect his portfolio.

The rendering styles that the user can select currently include audio, pager, and visual. The notification component 138 will incorporate other types of rendering mechanisms as they become viable. The user may also select a pre-notification

notification with the audio rendering style. If this pre-notification notification is selected, the notification component 138 alerts the user that an audio notification is arriving. The alert allows the user to listen for the incoming notification. Without the alert, the user may misinterpret or not hear the first few words of the notification because the user is

5 focusing on running a task. The notification component 138 renders the audio style by converting the notification text into audio and using the speakers 197. If speakers are not present, the notification component 138 renders the notification using the visual rendering style. In one embodiment, the notification component 138 is actionable (i.e., one or more actions are performed) when the user says a keyword or key-phrase in response to hearing

10 the notification. For example, if the notification is a financial notification that is about the user's mutual fund performance and the user says the keyword or key-phrase (e.g., "show me"), then the notification component 138 performs a specified function (e.g., brings up the web-site of the user's mutual fund). The pager rendering style pages the user as known by those skilled in the art.

15 The visual rendering style has several options. The notification can be rendered with a transparent display, an alpha-blended display, and a transparent alpha-blended display. The user specifies the font and font size of the notifications, the color of the notifications in the displays and where on the monitor 191 that the notification is to be rendered for each notification classification. Figure 4 illustrates exemplary display areas

20 304, 306, and 308 on display 300 of monitor 191 that the user may define and use to display visual notifications. Note that the user can define any area at any position on the display 300. The status bar 302 may also be used as a display area for visual notifications. The transparent display 310 (see figure 5a) and alpha-blended display 312,

09705658.1.0300

5

15

20

notification sent to the notification component 138. An example of a notification providing both the short version and the long version in XML form is:

```
<notification>
```

```
  <gui text="MSFT +2 / 82 / volume up" shorttext="MSFT +2/82/up"/>
```

```
5    <spoken text="Microsoft up 2 points to 82 on increased volume"
```

```
      shorttext="Microsoft up 2 at 82" />
```

```
</notification>
```

If the applications programs 135 only provides one text string, then the notification component 138 renders the text string provided using the rendering style chosen.

10 The user also specifies the priority of how notifications are rendered. The priorities that are specified include which notification classifications have priority over other notification classifications and how often the notification component 138 should notify the user.

As previously indicated, the notification component 138 renders the notifications
15 in accordance with the notification classification and the user's preferences. Once a notification is received, the notification component 138 parses the notification to determine the notification classification and the notification to be rendered. If the user's preferences indicate that the notification classification is enabled and the particular application is not disabled, the notification is rendered in accordance with the rendering
20 style and priority specified in the user's preferences.

In some instances, a number of notifications to be rendered are present at the notification component 138. In these instances, notifications are queued by the notification component 138. There are two types of queues that the notification

09705858.110300

component 138 uses. The first type is a modified strict queue. Notifications in a strict queue are queued in the order they are received and are rendered in the same order. In the modified strict queue, the notifications are queued by priority and in the order they are received. They are then rendered in the same order (i.e., from highest priority to lowest priority in the order the notifications are received in each priority level) and may be stacked vertically in the display area. The second type of queue is a flushed queue. In the flushed queue, the queue is flushed of all messages when a new notification arrives. For example, if a compact disc player changes state or changes song, the queue is flushed of previous state changes or song changes because they are out of date.

In one embodiment, the notification component 138 has a history feature that keeps track of the notifications that have not been rendered. Notifications from the history may be flushed by the notification component 138 once they have been rendered to the user. Older notifications are also flushed from the history. The time period that the notifications are kept is selected by the user, and may be set for each notification classification. The user also may select how the notifications in the history are displayed. The history feature is also actionable (i.e., one or more actions are performed when the user selects a notification in the history).

A centralized notification system has been described that allows a user to continue working on the task at hand while receiving a notification. The system allows a user to set preferences globally and by notification classification, allows the user to enable or disable notification classifications, and allows the user to set priority levels of notifications. In view of the many possible embodiments to which the principles of this invention may be applied, it should be recognized that the embodiment described herein

with respect to the drawing figures is meant to be illustrative only and should not be taken as limiting the scope of invention. For example, those of skill in the art will recognize that the elements of the illustrated embodiment shown in software may be implemented in hardware and vice versa or that the illustrated embodiment can be modified in

- 5 arrangement and detail without departing from the spirit of the invention. Therefore, the invention as described herein contemplates all such embodiments as may come within the scope of the following claims and equivalents thereof.

09705858 110300